



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
|-----------------|-------------|----------------------|---------------------|------------------|

10/662,034

09/12/2003

Paul J. Mantey

200309970-1

9298

22879 7590 04/06/2007

HEWLETT PACKARD COMPANY  
P O BOX 272400, 3404 E. HARMONY ROAD  
INTELLECTUAL PROPERTY ADMINISTRATION  
FORT COLLINS, CO 80527-2400

EXAMINER

SPITTLE, MATTHEW D

ART UNIT

PAPER NUMBER

2111

| SHORTENED STATUTORY PERIOD OF RESPONSE | MAIL DATE | DELIVERY MODE |
|--|-----------|---------------|
|--|-----------|---------------|

3 MONTHS

04/06/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

**Office Action Summary**

Application No.

10/662,034

Applicant(s)

MANTEY ET AL.

Examiner

Matthew D. Spittle

Art Unit

2111

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 09 February 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-13 and 42-44 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-13 and 42-44 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

Claims 1 – 13, and 42 – 44 have been examined.

#### ***Claim Rejections - 35 USC § 102***

5           The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

10           (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1 – 4, 12 and 43 are rejected under 35 U.S.C. 102(b) as being anticipated by Johnson et al. (U.S. 6,122,758).

Regarding claim 1, Johnson et al. describe a computer system comprising:

15           A system bus implemented in accordance with an Inter-IC bus specification (Figure 4, 7, item 310; column 7, lines 10 – 12);

A bus controller coupled to the system bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65) and to an internal bus (Fig. 4, 7, item 226);

20           A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32);

A first first-in first-out (FIFO) buffer coupled to the send machine, the first FIFO further coupled between the host processor and the bus controller (where a FIFO buffer  
25           may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 – 10)

over an internal bus (Figure 7, item 226) but not over the system bus, the first FIFO not being coupled to the send machine over the internal bus (Examiner notes that FIFO (516) is connected to the send machine (707) over intermediate DATA bus and not over the internal bus (226)).

30

Regarding claim 2, Johnson et al. describe the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but  
35 note that the interrupt is not generated until a response is received from the I2C device).

Regarding claim 3, Johnson et al. describe wherein:

The first FIFO buffer comprises means for receiving a plurality of bytes from the  
40 host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes);

The send machine comprises means for transmitting the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 – 38, and  
45 column 15, lines 40 - 51 tell of how a messages is placed into the request buffer and processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

Regarding claim 4, Johnson et al. describe:

50        A receive machine (Figure 7, item 707) coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32). Examiner notes that the MDR functions as both a send and receive machine, as described in column 14, lines 28 – 31).

55        A second FIFO buffer coupled to the receive machine and coupled between the host processor and the bus controller (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14).

Regarding claim 12, Johnson et al. teach a computer system comprising:

60        A system bus implemented in accordance with an Inter-IC bus specification (Figure 4, 7, item 310; column 7, lines 10 - 12);

      A bus controller coupled to the system bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65) and to a first internal bus (Fig. 7, 226) and a second internal bus (interpreted as the “short” bus that connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES);

65        A send machine (Figure 7, item 707; where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 – 32) coupled between a host processor (Figure 4, item 200) and the bus controller (Figure 7, item 312), the send machine comprising means for transmitting the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines

70 40 - 51 tell of how a messages is placed into the request buffer and processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

A first first-in first-out (FIFO) buffer (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 – 10) coupled to the send  
75 machine (Figure 7, item 707) and coupled between the host processor (Figure 4, item 200) and the bus controller (Figure 7, item 312) over a first internal bus (Figure 7, item 226) but not over the system bus (Figure 7, item 310), the first FIFO not being coupled to the send machine over the first internal bus, the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host  
80 processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707; where a receive machine may be  
85 interpreted as a message data register (MDR); column 12, lines 17 – 32) coupled between the host processor (Figure 4, item 200) and the bus controller, the receive machine comprising means for receiving the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed  
90 to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the

reference teaches the limitation of receiving the plurality of bytes without interrupting the processor). Examiner notes that the MDR functions as both a send and receive machine, as described in column 14, lines 28 – 31).

95           A second FIFO buffer (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 – 14) coupled to the receive machine (Figure 7, item 707), the second FIFO further coupled between the host processor and the bus controller over a second internal bus (interpreted as the “short” bus that connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the second  
100   FIFO not being coupled to the receive machine over the second internal bus but not over the system bus, the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted.  
105   Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

Regarding claim 43, Johnson et al. teach a computer system comprising:

110           A system bus (Figure 4, 7, item 310; column 7, lines 10 - 12);

          A bus controller coupled to the system bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 – 36, 61 - 65) and to a first internal bus (Fig. 4, 7, item 226) and a second internal bus (interpreted as the

"short" bus that connects from the SYSTEM INTERFACE PROCESSOR to the  
115 QUEUES);

A send machine (Figure 7, item 707; where a send machine may be interpreted  
as a message data register (MDR); column 12, lines 17 – 32) coupled between a host  
processor (Figure 4, item 200) and the bus controller (Figure 7, item 312), the send  
machine comprising means for transmitting the plurality of bytes over the system bus  
120 without interrupting the host processor (column 14, lines 18 – 38, and column 15, lines  
40 - 51 tell of how a messages is placed into the request buffer and processed by the  
interface, but note that the interrupt may not be generated until a response is received  
from the I2C device).

A first first-in first-out (FIFO) buffer (where a FIFO buffer may be interpreted as a  
125 request queue; Figure 7, item 516; column 12, lines 7 – 10) coupled to the send  
machine (Figure 7, item 707) and coupled between the host processor (Figure 4, item  
200) and the bus controller (Figure 7, item 312) over a first internal bus (Figure 7, item  
226) but not over the system bus (Figure 7, item 310), the first FIFO not being coupled  
to the send machine over the first internal bus, the first FIFO comprising means for  
130 receiving a plurality of bytes from the host processor without interrupting the host  
processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues  
are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes;  
column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but  
note that the interrupt is not generated until a response is received from the I2C device).



135           A receive machine (Figure 7, item 707; where a receive machine may be  
interpreted as a message data register (MDR); column 12, lines 17 – 32) coupled  
between the host processor (Figure 4, item 200) and the bus controller, the receive  
machine comprising means for receiving the plurality of bytes over the system bus  
without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 –  
140 51 tell of how the response bytes are written into the response FIFO, and the if directed  
to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean  
that the device driver may also *not* be directed to assert an interrupt, and therefore the  
reference teaches the limitation of receiving the plurality of bytes without interrupting the  
processor). Examiner notes that the MDR functions as both a send and receive  
145 machine, as described in column 14, lines 28 – 31).

          A second FIFO buffer (where a FIFO buffer may be interpreted as a response  
queue; Figure 7, item 514; column 12, lines 10 –14) coupled to the receive machine  
(Figure 7, item 707), the second FIFO further coupled between the host processor and  
the bus controller over a second internal bus (interpreted as the “short” bus that  
150 connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the second  
FIFO not being coupled to the receive machine over the second internal bus but not  
over the system bus, the second FIFO buffer comprising means for receiving a plurality  
of bytes from the bus controller without interrupting the host processor (column 14, lines  
18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the  
155 response FIFO, and the if directed to do so by the device driver, an interrupt is asserted.  
Examiner interprets this to mean that the device driver may also *not* be directed to

Art Unit: 2111

assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

160

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

165

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

170

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

175

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

180

Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. (U.S. 6,122,758) in view of Yoshida (U.S. 5,928,372).

Johnson et al. fail to teach wherein the receive machine comprises checksum generation means for generating a message checksum for a message while the message is being received by the bus controller over the system bus.

185

Yoshida teaches a checksum generation means for generating a message checksum for a message while the message is being received by the bus controller over the system bus (where checksum generation means may be interpreted as data check code generation circuits; column 11, lines 14 – 24).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to combine the checksum generation means of Yoshida with the system of Johnson et al. in order to provide for a means of verifying the data transmitted across the system bus. This would have been obvious since error-free data is critical to the correct operation of a digital system.

\* \* \*

Claims 6 – 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Feeney et al. (U.S. 6,072,781).

With regard to claim 6, Johnson et al. describe the computer system of claim 1, further comprising:

Means for receiving a message from the host processor (Figure 7, items 516, 707; column 12, lines 26 – 32);

Means for attempting to send the message over the system bus to a target device (column 15, lines 15 – 36 give an example of how a message is sent over the system bus to a target device (microcontroller)).

Means for determining whether the message was received without errors by the target device (column 15, lines 62 – 64).

Johnson et al. fail to describe retry means for attempting again to send the message over the communication bus to the target device if it is determined that the message was not received without errors by the target.

210           Feeney et al. teach retry means for attempting again to send the message over the system bus to the target device without interrupting the host processor if it is determined that the message was not received without errors by the target device (column 16, lines 36 – 49 describe retrying messages that failed to send; column 16, lines 36 – 49 describe retrying a message without involving the processor).

215           It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the means of retrying failed messages as taught by Feeney et al. into the computer system of Johnson et al for the purpose of ensuring the delivery of messages on the communication bus.

220           With regard to claim 7, Feeney et al. teach the additional limitation wherein the retry means comprises means for attempting again to send the message over the system bus to the target device without interrupting the host processor if it is determined that the message was not received without errors by the target device (column 16, lines 36 – 49 describe retrying a message without involving the processor).

225           With regard to claim 8, Feeney et al. teach the additional limitation wherein the retry means comprises means for attempting again to send the message over the system bus to the target device without obtaining the message again from the host processor if it is determined that the message was not received without errors by the target device (column 16, lines 36 – 49 describe storing the message in a FIFO in order to allow the processor to move onto other tasks).

230

\* \* \*

235 Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson  
et al. in view of Cao et al. (U.S. 5,230,044).

Johnson et al. fail to teach a busfree count means for storing a busfree count  
associated with the computer system, a busfree timer for use by the computer system to  
wait an amount of time specified by the busfree count prior to attempting to access the  
240 system bus after the system bus becomes available for use, and a fair arbitration block  
coupled between the host processor and the bus controller, the fair arbitration block  
comprising arbitration means for modifying the busfree count according to a priority  
signal to produce an arbitrated busfree count signal.

Cao et al. teach:

245 A busfree count means for storing a busfree count (where a busfree count may  
be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time  
specified by the busfree count prior to attempting to access the system bus after the  
system bus becomes available for use (where a busfree timer may be interpreted as a  
250 "quiet slot" counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree  
count according to a priority signal to produce an arbitrated busfree count signal (where

Art Unit: 2111

a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

255           It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Johnson et al. for the purpose of providing arbitration amongst devices on the system bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwidth (column 10, 260 lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

\*       \*       \*

265           Claims 10 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Webb et al. (U.S.4,577,060).

With regard to claim 10, Johnson et al. fail to teach a byte timer coupled between the bus controller and the host processor.

Webb et al. teach a byte timer (where a byte timer may be interpreted as a no- 270 response timer; column 13, lines 49 - 60).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to include the byte timer as taught by Webb et al. into the system of Johnson et al. This would have been obvious in order to provide a method of ensuring that a communication link (or bus) is operating properly, and prevent the

275 system from wasting time sending messages to processors/terminals that are not  
responsive (column 14, lines 11 – 19).

With regard to claim 11, Webb et al. teach the additional limitation wherein the  
byte timer (interpreted as a no-response timer) comprises means for determining  
280 whether the host processor has failed and means for generating a signal indicating  
whether the host processor has failed (where a host processor may be interpreted as a  
terminal; where generating a signal indicating the processor has failed may be  
interpreted as marking a terminal as being “offline” or “down”; column 13, line 49 –  
column 14, line 30).

285

\* \* \*

Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson  
et al. in view of Feeney et al., Cao et al., and further in view of Webb et al.

290 Johnson et al. teach a computer system of claim 12 further comprising:

Means for receiving a message from the host processor (Figure 7, items 516,  
707; column 12, lines 26 – 32);

Means for attempting to send the message over the system bus to a target  
device (column 15, lines 15 – 36 give an example of how a message is sent over the  
295 system bus to a target device (microcontroller)).



Means for determining whether the message was received without errors by the target device (column 15, lines 62 – 64).

Johnson et al. fail to describe a retry means, a busfree count means, a busfree count timer, a fair arbitration block, and a byte timer.

300 Feeney et al. teach retry means for attempting again to send the message over the system bus to the target device without interrupting the host processor if it is determined that the message was not received without errors by the target device (column 16, lines 36 – 49 describe retrying messages that failed to send; column 16, lines 36 – 49 describe retrying a message without involving the processor).

305 It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the means of retrying failed messages as taught by Feeney et al. into the computer system of Liu et al for the purpose of ensuring the delivery of messages on the communication bus.

Cao et al. teach:

310 A busfree count means for storing a busfree count (where a busfree count may be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the system bus after the system bus becomes available for use (where a busfree timer may be interpreted as a  
315 “quiet slot” counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where

Art Unit: 2111

a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

320           It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Johnson et al., Elliot, and Feeney et al, for the purpose of providing arbitration amongst devices on the system bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus  
325   bandwidth (column 10, lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

          Webb et al. teach a byte timer comprising means for determining whether the host processor has failed and means for generating a signal indicating whether the host processor has failed (where a byte timer may be interpreted as a no-response timer;  
330   column 13, lines 49 – 60; where a host processor may be interpreted as a terminal; where generating a signal indicating the processor has failed may be interpreted as marking a terminal as being “offline or “down”; column 13, line 49 – column 14, line 30).

          It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to include the byte timer as taught by Webb et al. into the system  
335   of Johnson et al., Feeney et al., and Cao et al. This would have been obvious in order to provide a method of ensuring that a communication link (or bus) is operating properly, and prevent the system from wasting time sending messages to processors/terminals that are not responsive (column 14, lines 11 – 19).

340

\* \* \*

Claim 42 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. in view of Feeney et al., Cao et al., and Webb et al.

Regarding claim 42, Johnson et al. teach a computer system comprising:

345

A system bus implemented in accordance with an Inter-IC bus specification (Figure 4, 7, item 310; column 7, lines 10 - 12);

A bus controller coupled to the system bus (where a bus controller may be interpreted as a system interface processor; column 11, lines 31 - 36, 61 - 65) and to an internal bus (Fig. 4, 7, item 226);

350

A send machine (Figure 7, item 707) coupled between a host processor (Figure 4, item 200) and the bus controller (where a send machine may be interpreted as a message data register (MDR); column 12, lines 17 - 32), the send machine comprising means for transmitting the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 - 38, and column 15, lines 40 - 51 tell of how a  
355 messages is placed into the request buffer and processed by the interface, but note that the interrupt may not be generated until a response is received from the I2C device).

A first first-in first-out (FIFO) buffer (where a FIFO buffer may be interpreted as a request queue; Figure 7, item 516; column 12, lines 7 - 10) coupled to the send machine (Figure 7, item 707) and coupled between the host processor (Figure 4, item  
360 200) and the bus controller (Figure 7, item 312) over a first internal bus (Figure 7, item 226) but not over the system bus (Figure 7, item 310), the first FIFO not being coupled

Art Unit: 2111

to the send machine over the first internal bus, the first FIFO comprising means for receiving a plurality of bytes from the host processor without interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show that both queues  
365 are a size of 1K – either bits or bytes would meet the limitation of a plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the request buffer, but note that the interrupt is not generated until a response is received from the I2C device).

A receive machine (Figure 7, item 707) coupled between the host processor and the bus controller (where a receive machine may be interpreted as a message data  
370 register (MDR); column 12, lines 17 – 32), the receive machine comprising means for receiving the plurality of bytes over the system bus without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted. Examiner interprets this to mean that the device driver  
375 may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

A second FIFO buffer (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 – 14) coupled to the receive machine (Figure 7, item 707), the second FIFO further coupled between the host processor and  
380 the bus controller over a second internal bus (interpreted as the “short” bus that connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the second FIFO not being coupled to the receive machine over the second internal bus but not over the system bus, the second FIFO buffer comprising means for receiving a plurality

Art Unit: 2111

of bytes from the bus controller without interrupting the host processor (column 14, lines  
385 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the  
response FIFO, and the if directed to do so by the device driver, an interrupt is asserted.  
Examiner interprets this to mean that the device driver may also *not* be directed to  
assert an interrupt, and therefore the reference teaches the limitation of receiving the  
plurality of bytes without interrupting the processor).

390 Means for receiving a message from the host processor (Figure 7, items 516,  
707; column 12, lines 26 – 32);

Means for attempting to send the message over the system bus to a target  
device (column 15, lines 15 – 36 give an example of how a message is sent over the  
system bus to a target device (microcontroller)).

395 Means for determining whether the message was received without errors by the  
target device (column 15, lines 62 – 64).

Johnson et al. fail to describe a retry means, a busfree count means, a busfree  
count timer, a fair arbitration block, and a byte timer.

Feeney et al. teach retry means for attempting again to send the message over  
400 the system bus to the target device without interrupting the host processor if it is  
determined that the message was not received without errors by the target device  
(column 16, lines 36 – 49 describe retrying messages that failed to send; column 16,  
lines 36 – 49 describe retrying a message without involving the processor).

It would have been obvious to one of ordinary skill in this art at the time of  
405 invention by applicant to incorporate the means of retrying failed messages as taught by

Art Unit: 2111

Feeney et al. into the computer system of Johnson et al. for the purpose of ensuring the delivery of messages on the communication bus.

Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may  
410 be interpreted as an arbitration count number; column 4, lines 42 – 50);

A busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the system bus after the system bus becomes available for use (where a busfree timer may be interpreted as a “quiet slot” counter; column 4, lines 51 – 60);

415 A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of  
420 invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Johnson et al. for the purpose of providing arbitration amongst devices on the system bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwidth (column 10, lines 25 – 59), along with permitting data communication with a very small probability of  
425 data collisions (column 4, lines 32 – 34).

Webb et al. teach a byte timer comprising means for determining whether the host processor has failed and means for generating a signal indicating whether the host

Art Unit: 2111

processor has failed (where a byte timer may be interpreted as a no-response timer;  
column 13, lines 49 – 60; where a host processor may be interpreted as a terminal;

430 where generating a signal indicating the processor has failed may be interpreted as  
marking a terminal as being “offline or “down”; column 13, line 49 – column 14, line 30).

It would have been obvious to one of ordinary skill in this art at the time of  
invention by applicant to include the byte timer as taught by Webb et al. into the system  
of Johnson et al. This would have been obvious in order to provide a method of  
435 ensuring that a communication link (or bus) is operating properly, and prevent the  
system from wasting time sending messages to processors/terminals that are not  
responsive (column 14, lines 11 – 19).

\* \* \*

440

Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson  
et al. in view of Cao et al.

Regarding claim 44, Johnson et al. teach a device for use in a computer system  
including a system bus (Figure 4, 7, item 310; column 7, lines 10 - 12) and a bus  
445 controller coupled to the system bus (where a bus controller may be interpreted as a  
system interface processor; column 11, lines 31 – 36, 61 - 65) and to a first internal bus  
(Fig. 4, 7, item 226) and a second internal bus (interpreted as the “short” bus that  
connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the device  
comprising:

450           A send machine (Figure 7, item 707) coupled between a host processor (Figure  
4, item 200) and the bus controller (where a send machine may be interpreted as a  
message data register (MDR); column 12, lines 17 – 32), the send machine comprising  
means for transmitting the plurality of bytes over the system bus without interrupting the  
host processor (column 14, lines 18 – 38, and column 15, lines 40 - 51 tell of how a  
455   messages is placed into the request buffer and processed by the interface, but note that  
the interrupt may not be generated until a response is received from the I2C device).

          A first first-in first-out (FIFO) buffer coupled to the send machine and coupled  
between the host processor and the bus controller (where a FIFO buffer may be  
interpreted as a request queue; Figure 7, item 516; column 12, lines 7 - 10, the first  
460   FIFO comprising means for receiving a plurality of bytes from the host processor without  
interrupting the host processor (column 12, lines 26 - 30; Figure 7, items 516, 514 show  
that both queues are a size of 1K – either bits or bytes would meet the limitation of a  
plurality of bytes; column 15, lines 40 – 51 tell of how a messages is placed into the  
request buffer, but note that the interrupt is not generated until a response is received  
465   from the I2C device).

          A receive machine (Figure 7, item 707) coupled between the host processor and  
the bus controller (where a receive machine may be interpreted as a message data  
register (MDR); column 12, lines 17 – 32), the receive machine comprising means for  
receiving the plurality of bytes over the system bus without interrupting the host  
470   processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response  
bytes are written into the response FIFO, and the if directed to do so by the device



driver, an interrupt is asserted. Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

475           A second FIFO buffer (where a FIFO buffer may be interpreted as a response queue; Figure 7, item 514; column 12, lines 10 –14) coupled to the receive machine (Figure 7, item 707), the second FIFO further coupled between the host processor and the bus controller over a second internal bus (interpreted as the “short” bus that connects from the SYSTEM INTERFACE PROCESSOR to the QUEUES), the second  
480   FIFO not being coupled to the receive machine over the second internal bus but not over the system bus, the second FIFO buffer comprising means for receiving a plurality of bytes from the bus controller without interrupting the host processor (column 14, lines 18 – 38; column 15, lines 40 – 51 tell of how the response bytes are written into the response FIFO, and the if directed to do so by the device driver, an interrupt is asserted.  
485   Examiner interprets this to mean that the device driver may also *not* be directed to assert an interrupt, and therefore the reference teaches the limitation of receiving the plurality of bytes without interrupting the processor).

Johnson et al. fail to describe a busfree count means, a busfree timer, and a fair arbitration block.

490           Cao et al. teach:

A busfree count means for storing a busfree count (where a busfree count may be interpreted as an arbitration count number; column 4, lines 42 – 50);

Art Unit: 2111

A busfree timer for use by the computer system to wait an amount of time specified by the busfree count prior to attempting to access the system bus after the system bus becomes available for use (where a busfree timer may be interpreted as a "quiet slot" counter; column 4, lines 51 – 60);

A fair arbitration block comprising arbitration means for modifying the busfree count according to a priority signal to produce an arbitrated busfree count signal (where a busfree count may be interpreted as an arbitration count number; column 5, lines 59 – 64).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to incorporate the busfree count and busfree timer as taught by Cao et al. into the system of Johnson et al. for the purpose of providing arbitration amongst devices on the system bus. This would have been obvious since Cao et al. teach that their invention provides for more efficient use of bus bandwidth (column 10, lines 25 – 59), along with permitting data communication with a very small probability of data collisions (column 4, lines 32 – 34).

### ***Response to Arguments***

Applicant's arguments filed 2/9/2007 have been fully considered but they are not persuasive.

Regarding Applicant's argument that the bus controller (312) in Fig. 7 of Johnson is not coupled to the internal bus (226), Examiner points to Fig. 4, which clearly shows the bus controller (312) coupled to the internal bus (226). The intervening elements

Art Unit: 2111

515 disclosed in Fig. 7 are not shown in Fig. 4, however, the claim language does not limit intervening elements from being present, and more specifically, limit the bus controller from being coupled to the internal bus *through* those intervening elements.

Examiner would like to point out that the distinction between whether two devices are “coupled” or “directly coupled” is necessary in this art, as evidenced by Yoo et al.

520 (U.S. 6,834,014), in column 3, lines 41 – 48.

### ***Conclusion***

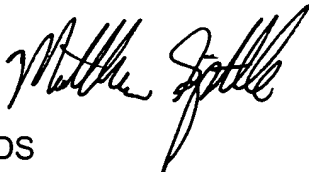
Any inquiry concerning this communication or earlier system from the examiner should be directed to Matthew D. Spittle whose telephone number is (571) 272-2467.


525 The examiner can normally be reached on Monday - Friday, 8 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Mark Rinehart can be reached on 571-272-3632. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

530 Patent Application Information Retrieval (PAIR) system. Status information for  
published applications may be obtained from either Private PAIR or Public PAIR.  
Status information for unpublished applications is available through Private PAIR only.  
For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should  
you have questions on access to the Private PAIR system, contact the Electronic  
535 Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a  
USPTO Customer Service Representative or access to the automated information  
system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

540 MDS 

  
**MARK H. RINEHART**  
**SUPERVISORY PATENT EXAMINER**  
**TECHNOLOGY CENTER 2100**